Natural landmark localisation for RoboCup

Peter Anderson (3293464) and Yongki Yusmanthia (3269084); Mentor Dr. Bernhard Hengst

***Abstract*—This paper outlines a fast and effective method for mapping natural landmarks to improve localisation during RoboCup soccer matches. The method uses modified 1D SURF features extracted from the row of pixels on the robot's horizon. These features are then matched across images using a dynamic programming algorithm that ensures that features can only be matched if they appear in the same order in each image. We demonstrate that this approach is robust to lighting changes, occlusion and small changes in viewing angle, and computationally cheap enough to run on the Nao humanoid robot in real time. As such we are confident that with further work this approach can be used as the basis of an enhanced localisation system for playing robot soccer.**

***Index Terms*—Feature extraction, Humanoid robots, Robot vision systems, Simultaneous localisation and mapping**

I. Introduction

THIS paper outlines a fast and effective method for extracting and matching natural landmarks, designed to improve robot localisation in the RoboCup Standard Platform League (SPL). The RoboCup SPL is an international robot soccer league based on the Aldebaran Nao, a humanoid robot equipped with a 500MHz Geode LX800 processor and a 640 x 480 pixel 30 fps digital camera.

Currently the rUNSWift team uses field lines and goal-posts for localisation, and images of anything beyond the field are ignored. This approach has been successful because the goal-posts at each end of the field have been painted in contrasting colours, enabling each half of the field to be distinguished from the other. However, from 2012 the goal-posts are expected to be uniformly coloured, meaning that field-ends will be aliased unless existing localisation techniques are extended. The motivation for this project is therefore to use some natural landmarks beyond the field to improve localisation. If goal colours are removed this will be an essential development.

Although there are several effective approaches to extracting landmarks or features from images [1], [2] the requirement for real-time soccer playing on the Nao imposes significant limitations on the computational expense that can be spared. With the competing demands of other robot vision, localisation and movement behaviours, as well as the 30 fps frame rate, we anticipate that any method of natural landmark localisation should ideally be constrained to 5 - 10 milliseconds of execution time at most.

Given this processing constraint, our method for natural landmark detection and description uses a 1 dimensional (1D) modified SURF algorithm [1] applied to a single row of grey-scale pixels. To enable accurate and efficient matching of features across images, we consider 3 matching algorithms, including nearest neighbour, nearest neighbour with ordering constraint, and nearest neighbour with ordering and scaling constraints. The last of these techniques was found to be the most accurate. This technique is implemented using a dynamic programming algorithm with $O(n^2)$ time complexity, which is equivalent to a naïve implementation of nearest neighbour. Overall, our approach is most similar to [5], who use a 1D variant of SIFT and a dynamic programming matching algorithm to localise a robot using an omni-directional camera. However, our approach differs in some important respects since we use SURF rather than SIFT for feature detection and extraction, and because the Nao camera has a viewing angle of only 40 degrees rather than 360 degrees.

II. Related Research

There is a considerable body of previous research on identifying feature representations that are stable under scale and viewpoint changes. Scale-Invariant Feature Transform (SIFT) [2] transforms image data into scale-invariant coordinates relative to local features. Key features in SIFT are defined as maxima and minima of the result of the difference of Gaussians function applied in scale-space to a series of smoothed and resampled images. Low contrast candidate features and edge response points along an edge are discarded.

Another popular feature detection algorithm is Speeded Up Robust Features (SURF) [1] which is related to SIFT. Instead of using a Difference of Gaussian filter, SURF makes an efficient use of box filters which approximates second order Gaussian derivatives and can be evaluated very fast using integral images. Using a standard testing procedure to match features across images, [1] found SURF to be faster and more accurate than SIFT. On average it finished the task in 354 ms and was correct 82.6% of the time compared to SIFT which finished in 1036 ms and was correct 78.1% of the time.

Previous research undertaken by the rUNSWift team [4] into the natural landmarks localization problem ruled out feature detection methods as computationally too expensive, and investigated matching of horizon pixels using Region Binning and Cross-Correlation. In each case the 360-degree horizon ring of pixels was sampled from the centre of the Robocup field and stitched together to obtain a stored ring of pixels. Horizon pixels extracted from subsequent images taken at random headings and locations around the field were compared with the stored ring of pixels, to try to determine a likelihood distribution over the heading of the robot.

Splitting the horizon ring of pixels into 28 bins, Region Binning and Cross-Correlation produced a likelihood maximum in the correct bin with probability 22% and 46% respectively. It was concluded that while both techniques showed some promise, they were far from practical use in the RoboCup competition due to the length of execution time, and lack of robustness in the face of changes in scale and viewpoint.

The closest work to ours is [5],[11],[12], which applies SIFT to a 1D circular panoramic image and uses both

colour information and curvature as the feature descriptor for robot navigation. Features are matched using a dynamic programming algorithm which in this case is circular. 1D SIFT is also used by [6] as a shape-based, time-scale invariant feature descriptor for 1D sensor signals. However, we decided to use SURF rather than SIFT as our starting point since [7] document that SIFT is faster than SURF for a range of real-world images and offers comparable matching performance.

III. Project Goals and Scope

A. Primary Goal

The primary goal of this project is to establish and implement a fast, robust algorithm for detecting and extracting features from images of typical RoboCup field surroundings (such as indoor office environments). The algorithm can make use of the known location of the horizon in the image, and any other known facts about the Robocup environment, but it must be fast enough to run in real time on a Nao. Furthermore, the extracted features need to be repeatable in the face of small viewpoint changes, horizon errors, image blurring, changes in image brightness and partial occlusion by other robots.

B. Secondary Goal

The secondary goal of this project is to implement a scalable method of matching extracted features across images. Whilst preprocessing can be used, the matching step must be fast enough to run in real time on a Nao, and ultimately scalable enough to be used in a real time RoboCup soccer match. We note that during a 20 minute RoboCup match the Nao will observe approximately 36,000 images using a 30 fps camera.

Fig. 1. Image taken from the Nao camera showing superimposed horizon line as determined by robot kinematics.

C. Scope

We are confident that the ability to extract and match repeatable image features (otherwise known as natural landmarks) in real time on the Nao will underpin a robust localisation system. Possible approaches to this localisation system include FastSLAM [8] or the Appearance-only SLAM system used by [3]. However, given the time constraints of this project we consider the implementation of the actual localisation system to be beyond the scope of this project.

IV. Problem Decomposition

Our approach to the project was to decompose the project into 4 stages. These stages consisted of: (1) implementing the 1DSURF feature extraction algorithm in C++, initially on a laptop; (2) implementing 3 different algorithms for matching features across images; (3) formally evaluating and tuning the system on a test bed of images; and (4) porting the optimised algorithm to the Nao robot and evaluating performance in real time.

Stage 1. Implementation of feature detection

In this stage we implemented and tested natural landmark detection and description using a 1D modified SURF algorithm [1] applied to a single row of image pixels. This row of pixels was extracted by taking a vertical average of 40 pixels at the robot's horizon (as determined by robot kinematics). The horizon pixels were chosen since objects on the horizon do not move vertically irrespective of the robots position on the field. We anticipated this approach would be several orders of magnitude faster than both SURF and Upright-SURF (U-SURF) as described in [1], which was reported to execute on a Linux 3GHz Pentium IV in 354 milliseconds and 255 milliseconds respectively using standard test images.

To rapidly prototype 1D SURF, we began by modifying the source code of the OpenSURF library [9]. This library was chosen since it is written in C++, the same language as the rest of the rUNSWift robot vision modules, and because it is open-source, self-contained and well documented.

Stage 2. Implementation of feature matching

For SURF features to be useful for localisation, we must be able to match the same features when they are observed in different images. Stage 2 therefore consisted of implementing several feature matching algorithms, include nearest neighbour, nearest neighbour with an ordering constraint, and nearest neighbour with both ordering and scaling constraints.

The ordering constraint attempts to improve matching accuracy by using the knowledge that all features are on the horizon line. Since the robot will never move behind any background features in RoboCup, this implies that the features visible from any two different locations should always have the same ordering. Similarly, the scaling constraint is the 1D equivalent of verifying that the matched features satisfy a valid homography.

As noted in [12], the dissimilarity of two features of the same type (both maxima or both minima) can be computed as the Euclidean distance between their feature vectors. Similarly to [12], we define the matching score $S_{i,j}$ of two features i and j to be the inverse of this distance, and the matching score of two images to be the sum of matching scores over all matched features.

In order to efficiently match features using the ordering constraint, we implemented a dynamic programming algorithm to find a set of feature matches that obeys the ordering constraint, while simultaneously maximizing the matching score of the two images. This algorithm is related to the method used by [12] to match circular panoramic images, but differs since our image is not circular. In pseudo-code, given two 1-indexed feature vectors of length m and n, the algorithm to return image matching score under the ordering constraint is as follows:

```
for i = 0, 1, 2,…, m:
   E(i,0) = 0
for j = 0, 1, 2,..., n:
   E(0,j) = 0
for i = 0, 1, 2,…, m:
   for j = 0, 1, 2,..., n:
      E(i,j) = max{ E(i-1,j), E(i,j-1), E(i-1,j-1)+S_{i,j} }
return E(m,n)
```

This algorithm is essentially a variation of the well known edit-distance algorithm. Horizontal and vertical steps in the E table correspond to leaving a feature in one vector or the other unmatched, whilst a diagonal step represents matching the next available features in each feature vector.

The method for matching under the ordering and scaling constraint is simply to match under the ordering constraint, as described above, and then use RANSAC [13] to exclude outliers from the line fitted to the points. Although this approach fails to allow for the curving of the matching line under robot translation, in practice we believe that with a camera viewing angle of only 40 degrees the curvature will be small.

Stage 3. Evaluation and tuning on a classification task

We originally intended to use the image sequences and testing software provided by [10] as used by [1] to evaluate the robustness of 1D SURF and tune various parameters. However, given the idiosyncrasies of the Nao robot and the potential for unknown error in the horizon location, we ultimately decided to evaluate the algorithm using images from the Nao robot.

To create a bank of test images, we captured 88 test images (and associated horizon location information) with the robot facing in all directions from a single location on the field. During this process the background around the field consisted of a typical office or computer laboratory environment.

In order to evaluate the accuracy of the feature extraction and matching algorithm, we generated a test bank of 480 matching image pairs (defined as two images with at least 50% overlap) and 2,065 unmatched image pairs (defined as two images with no overlap) from the original 88 test images. We then posed the localisation problem as a classification task, to see if the matching score between two images was a reliable indicator of whether two images actually contained the same landmarks.

Stage 4. Evaluation on the robot

The final stage of the project consisted of learning the rUNSWift architecture and porting the code to the Nao robot. We then conducting some more informal tests of the matching algorithm while actually running on the robot in real time. These tests included changes in scale, viewpoint, lighting conditions and removal of objects from the image.

V. Results from Classification of Test Images

Figures 2, 3 and 4 illustrate some typical output from our evaluation system, using the same two images from the test bank but with 3 different matching techniques. The scatter plot in the top right of the illustration depicts the features matched across the two images. The grey-scale row of pixels extracted from the horizon (itself marked in red) is shown above each image. The left-hand image has been flipped and rotated for the benefit of alignment with the scatter plot.

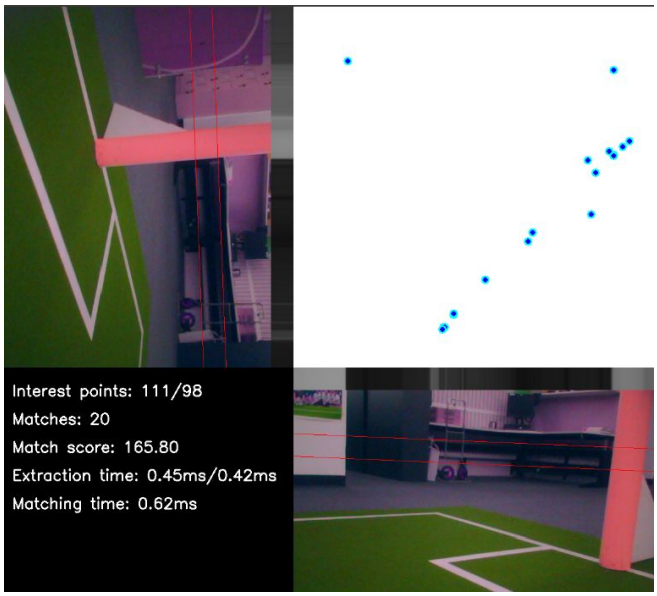

Fig. 2. Example of image matching using 1DSURF and nearest neighbour feature matching. Red lines show the extremities of the horizon region. This region is vertically averaged to extract the single row of grey pixels shown above each image. All 1DSURF features are extracted from this single row of grey pixels. Matches are shown as blue dots.

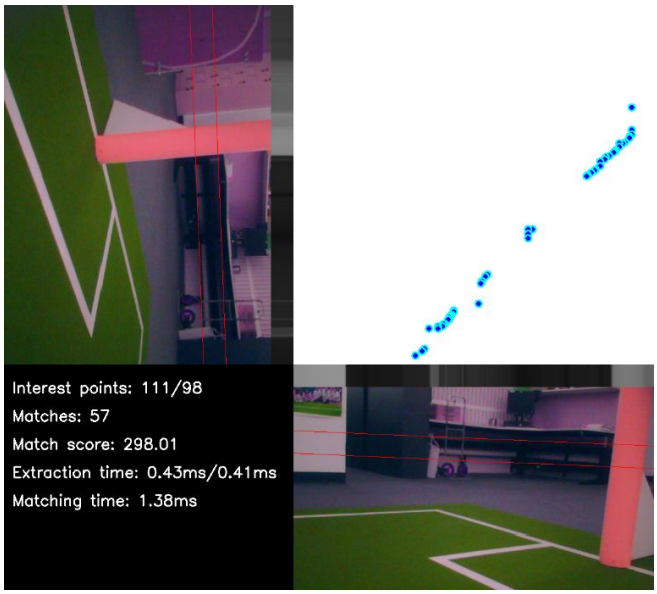

Fig. 3. Example of image matching using 1DSURF and feature matching with an Ordering constraint. This constraint ensures that features on the horizon must be matched in the same order in each image. This improves image matching at the cost of some additional processing time.

In terms of the performance of the 3 different matching techniques, Figure 2 clearly illustrates that nearest neighbour matching produces a number of incoherent matches, and also fails to match a large number of features that fail to satisfy the nearest neighbour distance ratio < 0.65 test. Adding the ordering constraint in Figure 3 both increases the number of correct matches and also discards some incorrect matches. Finally, the addition of the scaling constraint in Figure 4 manages to discard some additional incorrect matches.

A further advantage of using the scaling constraint is that finding the equation of the line through matching feature

points yields important visual odometry, or homography information. Suppose the line fitted to matching feature points is given by $x_2 = m*x_1 + b$, where x_2 is the pixel location in image 2, and x_1 is the pixel location in image 1. Assuming the horizon is close to horizontal in each image, the change in viewing angle θ from image 1 to image 2 (in pixels) is given by:

$$\theta = (m-1)*IMAGE\_WIDTH/2 + b$$

where IMAGE_WIDTH is the number of horizontal pixels in the image. For example, in Figure 4, the robot is estimated to have rotated 11.6 degrees to the right between the bottom image and the left hand image. In this case m=1.0 (no change in image scale), but in general the value of m gives important clues about the amount of movement toward or away from landmarks.

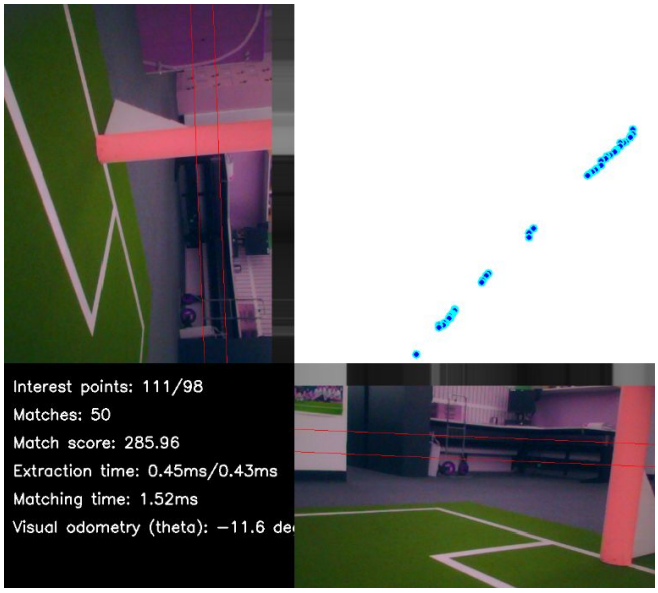

Fig. 4. Image matching using 1DSURF and feature matching using Ordering and Scaling constraints. Adding a further constraint to ensure that there is a consistent scale factor between features in each image improves image matching performance even further. In addition, using this matching technique the change in viewing angle can also be extracted. In this case the robot has rotated 11.6 degrees to the right between the bottom image and the top image.

Figure 5 depicts the overall performance of 1DSURF on the complete classification task described above in terms of a ROC curve. It can be seen that the addition of the ordering and scaling constraints result in a substantial improvement to classification accuracy. 1DSURF is clearly less robust than the original SURF (2DSURF) proposed by [1]. However, as illustrated in Table 1, our 1DSURF implementation extracts features nearly 500x faster than the OpenSURF [9] implementation when running on a typical laptop. With mean extraction time already below 0.5ms, we believe that with further profiling and tuning real-time feature extraction on the Nao during RoboCup matches should be feasible.

While in most cases the 1DSURF algorithm was implemented as the 1 dimensional analog of 2DSURF, after a substantial amount of experimentation some changes were made to suit the task at hand. For example, rather than selecting only the extrema in image scale-space as feature point locations, we ultimately required that points need only be extrema in the single space dimensional to be selected. This relaxation ensures that sufficient feature points will be detected.

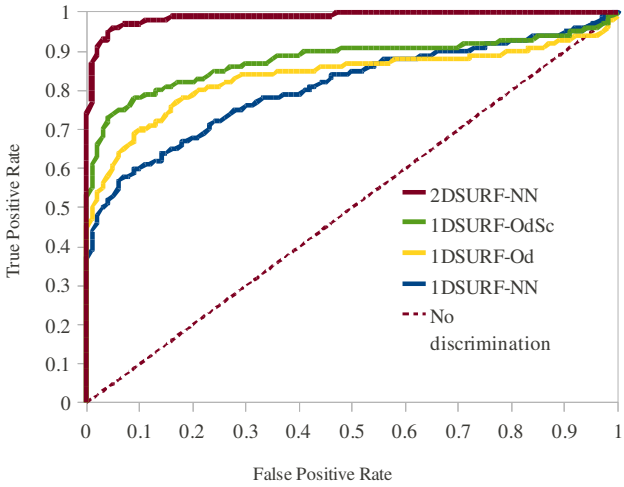

Fig. 5. ROC curve for classifying images as matched or unmatched using the matching score. 2DSURF has a better matching performance than any of our 1DSURF variations, but the running time is hundreds of times slower. OdSc = feature matching with Ordering and Scaling constraint, Od = feature matching with Ordering constraint only, NN = nearest neighbour matching with no constraint.

TABLE I. Running time of feature extraction and matching algorithms.

Feature extraction technique	Feature matching technique	Mean no. features	Mean extraction time (ms)	Mean matching time (ms)	AUC
2D SURF	Nearest neighbour (NN)	429	199.06	18.92	98.82%
1D SURF	Nearest neighbour (NN)	107.6	0.43	0.21	80.61%
1D SURF	NN with ordering	107.6	0.43	0.45	83.56%
1D SURF	NN with ordering & scaling	107.6	0.43	0.49	87.80%

AUC = Area Under ROC Curve. Times were evaluated on a 2.4GHz Core 2 Duo Processor laptop using a testing set consisted of 480 matching image pairs and 2065 unmatched image pairs (generated from 88 test images).

The original SURF interpolates the location of features in both space and scale to sub-pixel accuracy by fitting a 3D quadratic curve to the local image function. In our application, we found that the additional accuracy provided by this step was not worth the computational burden, and it was discarded. Similarly, the orientation assignment step of the original algorithm is not necessary, since the orientation of all 1DSURF points is specified by the horizon. Finally, although the 1DSURF feature descriptor is calculated analogously to the 2D version, due to the reduction in sample space it is an 8-dimensional vector rather than a 64-dimensional vector.

Following extensive experimentation, the following parameter values were chosen: 4 scale-space octaves of 3 intervals each, an initial sample scale of 2 pixels, a feature response threshold of 0.00005 and a horizon width of 40 pixels. Table II below provides an overview of the impact of changing the response threshold.

TABLE II. Impact of variations in the feature detection threshold on running time and accuracy, using 1D SURF with Ordering and Scaling constraints.

Feature detection threshold	Mean no. features	Mean extraction time (ms)	Mean matching time (ms)	AUC
0.5x	127.5	0.45	0.67	88.70%
1x	107.6	0.43	0.49	87.80%
2x	92.1	0.41	0.37	86.18%
4x	78.5	0.39	0.27	83.83%
8x	65.2	0.38	0.21	81.10%

AUC = Area Under ROC Curve. Times were evaluated on a 2.4GHz Core 2 Duo Processor laptop using a testing set consisted of 480 matching image pairs and 2065 unmatched image pairs (generated from 88 test images). Thresholds are relative to the chosen threshold of 0.00005.

VI. Results on the Nao Robot

As a general proof of concept, for the following sequence of results we have stored 10 background images taken from one side of the RoboCup field as a crude map. We then moved the robot to different locations around the field. We hoped to observe that the robot could recognise the same landmark when it was seen again, even with some translation, changes in lighting, and changes to the background itself. Comments on the results of each of these tests can be found in the figure captions. The matching algorithm used for these tests was nearest neighbours with the ordering and scaling constraint.

At all times the feature extraction algorithm was running in real-time on the robot. Currently this algorithm takes around 150ms to 200ms to execute on the robot (compared to less than 0.5ms on a typical laptop). While this is still too slow for the RoboCup competition, we are confident that with careful profiling and optimisation on the actual robot (which we haven't yet undertaken), further speed increases can be found.

As an aside, we note that in several cases the algorithm matches the test image to multiple map images. This should not be seen as a failure, since each correct match also provides homography information. As such, multiple correct matches should provide increased localistion accuracy.

1176 MATCH	**338 MATCH**	**228 MATCH**	88	77
32	62	78	61	51

Fig. 6. Successful matching of a near- identical image. The table depicts matching scores for the top image against the 10 images tiled in the lower frame. The threshold for images to be considered a match is 200.

254 MATCH	155	116	118	75
39	31	47	48	31

Fig. 7. Successful matching demonstrating robustness to changes in some background objects. In this case, even with the removal of the goal posts, the image is still correctly matched to the location where the goalposts were before. The table depicts matching scores for the top image against the 10 images tiled in the lower frame. The threshold for images to be considered a match is 200.

240 MATCH	175	132	100	89
41	30	44	51	26

Fig. 8. Successful matching with both changes to background objects (goal posts removed) and changing lighting conditions (field lights turned off). The table depicts matching scores for the top image against the 10 images tiled in the lower frame. The threshold for images to be considered a match is 200.

81	83	73	81	97
85	**275 MATCH**	**739 MATCH**	61	91

Fig. 9. Successful matching of a different base image. The table depicts matching scores for the top image against the 10 images tiled in the lowerframe. The threshold for images to be considered a match is 200.

110	150	**353 MATCH**	**710 MATCH**	**473 MATCH**
53	76	69	87	63

Fig. 10. Successful matching of another different base image. The table depicts matching scores for the top image against the 10 images tiled in the lower frame. The threshold for images to be considered a match is 200.

39	33	53	66	61
95	102	51	111	68

Fig. 11. Image is correctly rejected as not matching to any of the saved map images (it is in fact an image taken from the other side of the field) The table depicts matching scores for the top image against the 10 images tiled in the lower frame. The threshold for images to be considered a match is 200.

245 MATCH	199	**210 MATCH**	69	59
49	52	73	58	56

Fig. 12. Successful matching of the first image from a different angle (sideways translation of the robot to it's right). The table depicts matching scores for the top image against the 10 images tiled in the lower frame. The threshold for images to be considered a match is 200.

293 MATCH	**313 MATCH**	**238 MATCH**	101	115
99	79	58	83	40

Fig. 13. Successful matching of the first image with the robot placed closer to the goal posts. The table depicts matching scores for the top image against the 10 images tiled in the lower frame. The threshold for images to be considered a match is 200.

324 MATCH	**389 MATCH**	**311 MATCH**	122	101
29	64	62	35	45

Fig. 14. Successful matching of the first image with the robot placed further away from the goal posts. The table depicts matching scores for the top image against the 10 images tiled in the lower frame. The threshold for images to be considered a match is 200.

284 MATCH	150	110	69	99
37	72	86	67	53

Fig. 15. Successful matching of the first image from a different angle (sideways translation of the robot to its left). The table depicts matching scores for the top image against the 10 images tiled in the lower frame. The threshold for images to be considered a match is 200.

VII. Conclusions and Future Research Directions

The 1DSURF method and matching algorithm we have proposed has been shown to be accurate at detecting repeated natural landmarks in typical office or laboratory environments, when using the location of the horizon as determined by the robot. The algorithm is robust to changes in illumination, removal of some landmarks, small changes in viewing angle (sideways translation) and significant scale changes (translation forwards and backwards). In addition, the algorithm is fast enough to run in real-time on the Nao robot, although further speedups are required for it to be match ready.

Although we consider the current implementation as a proof of concept, there remains a considerable amount of future development to be undertaken. In terms of speeding up the algorithm, we would suggest detailed profiling and optimisation of the algorithm as running on the robot, which we haven't undertaken. For example, currently the algorithm sequentially accesses vertical pixels from horizon band. Depending on the layout of the image in memory, we suspect that sequentially accessing horizontal pixels might be faster, but we have not tested this idea. Furthermore, sub-sampling the original image has not been tested.

Secondly, we would suggest running the matching algorithm on sequential frames and using the extracted visual odometry information in the existing robot localisation system. We anticipate that this visual odometry information will prove to be highly accurate with regard to changes in robot rotation, which is not well estimated by the existing odometry from the robot walk engine.

Finally, before the system can be used for localisation, the speed of the matching algorithm needs to be increased. The existing algorithm using the ordering and scaling constraint is not scalable to large numbers of images. We see two possible approaches to this issue. One approach would be combining detected features into a single feature map, rather than matching over many previously stored images. This would involve a SLAM method, such as FastSLAM [8].

A second approach to the matching issue would be to implement a two-stage matching process, such that an approximate method is used to first narrow-down the list of likely image matches. The full matching algorithm with the ordering and scaling constraint could then be applied to the short list of likely candidates. An approach like this would be similar to [3]. In this paper, the authors implement a highly scalable system by mapping the continuous space of feature vectors into a discrete feature vocabulary. This vocabulary can be learned off-line by clustering features extracted from a set of training data, using the modified k-means clustering algorithm developed by [3]. The approximate matching step then consists of using an inverted index of dictionary features to efficiently index the previous images that contain those features. This short list of images can then be verified using the full matching algorithm.

VIII. Contributions of Individual Group Members and SubDivision of Tasks

Both authors were fully involved in the entire project. There was no sub-division of tasks.

References

[1] H. Bay, A. Ess, T. Tuytelaars and L. Van Gool, "SURF: Speeded Up Robust Features", *Computer Vision and Image Understanding (CVIU)*, Vol. 110, No. 3, 2008 pp. 346–359.

[2] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, January 2004.

[3] M. Cummins and P. Newman, "Appearance-only SLAM at large scale with FAB-MAP 2.0", *The International Journal of Robotics Research* 30:100, 2011.

[4] Y. Deng, "Natural Landmarks Localisation," unpublished, University of New South Wales, August 2011.

[5] A. J. Briggs, Y. Li, D. Scharstein, M. Wilder, "Robot Navigation using 1D Panoramic Images," *Proceedings of ICRA* 2006. pp. 2679-268

[6] J. Xie and M. S. Beigi, "A scale-invariant local descriptor for event recognition in 1D sensor signals", *Proceedings of ICME*. 2009, pp. 1226-1229

[7] L. Juan and O. Gwun, "A Comparison of SIFT, PCA-SIFT and SURF". *International Journal of Image Processing (IJIP)* Volume(3), Issue(4). 2009.

[8] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "Fastslam: A factored solution to the simultaneous localization and mapping problem. In AAAI-2002 (Vancouver, BC, July 2002).

[9] C. Evans. (2009, January). Notes on the OpenSURF Library. Available: http://www.chrisevansdev.com/computer-vision-opensurf.html

[10] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, L. Van Gool, "A comparison of affine region detectors," *International Journal of Computer Vision* 65, 2005, pp. 43-72.

[11] A. Briggs, C. Detweiler, P. Mullen, and D. Scharstein, "Scale-space features in 1D omnidirectional images," in *Omnivis 2004, the Fifth Workshop on Omnidirectional Vision*, May 2004, pp. 115–126.

[12] A. Briggs, Y. Li, and D. Scharstein, "Matching features across 1D panoramas," in *Omnivis 2005, the Sixth Workshop on Omnidirectional Vision*, Oct. 2005.

[13] Fischler, M., and Bolles, R. (1981). "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". *Comm. of the ACM* 24: 381–395.